Amendments to the Claims:

5. (Currently Amended) ~~The method of claim 4~~ A method for modeling a circuit design comprising:

synthesizing a circuit design to create a first gate-level representation of the circuit design;

analyzing a second gate-level representation of the circuit design to learn architecture information; and

resynthesizing the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

wherein the analyzing comprises a resource sharing learning,

~~wherein~~ the resource sharing learning comprising:

creating one or more resource pairs from sharable resources in the first gate-level representation of the circuit design;

for each of the one or more resource pairs, synthesizing a subcircuit that shares the resource pair;

for each of the synthesized subcircuits, calculating a similarity with a corresponding subcircuit in the second gate-level representation of the circuit design; and

identifying the synthesized subcircuits having a high similarity with the corresponding subcircuit in the second gate-level representation of the circuit design.

6. (Currently Amended) ~~The method of Claim 1~~ A method for modeling a circuit design comprising:

synthesizing a circuit design to create a first gate-level representation of the circuit design;

analyzing a second gate-level representation of the circuit design to learn architecture information; and

resynthesizing the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design, wherein the analyzing comprises an operator order learning.

7. (Previously Presented) The method of Claim 6, wherein the operator order learning comprising:

creating one or more operand pairs from operations in the first gate-level representation of the circuit design;

for each of the one or more operand pairs, synthesizing a subcircuit for the operand pair;

for each of the synthesized subcircuits, calculating a similarity with a corresponding subcircuit in the second gate-level representation of the circuit design; and

identifying the synthesized subcircuits having a high similarity with the corresponding subcircuit in the second gate-level representation of the circuit design.

8. (Original) The method of Claim 7 further comprising, for each of the synthesized subcircuits having high similarity, creating a new operand that signifies the output of the operand pair associated with the synthesized subcircuit.

9. (Currently Amended) ~~The method of Claim 1~~ A method for modeling a circuit design comprising:

synthesizing a circuit design to create a first gate-level representation of the circuit design;

analyzing a second gate-level representation of the circuit design to learn architecture information; and

resynthesizing the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

wherein the analyzing comprises a multiplier learning.

10. (Previously Presented) The method of Claim 9, wherein the multiplier learning comprising:

identifying a multiplier in the first gate-level representation of the circuit design;

identifying a corresponding multiplier subcircuit in the second gate-level representation of the circuit design; and synthesizing the multiplier in the first gate-level representation with a partial product generation implementation; and

calculating a similarity for the synthesized partial product generation subcircuit with the partial product generation subcircuit in the corresponding multiplier in the second gate-level representation.

11. (Previously Presented) The method of Claim 9, wherein the multiplier learning comprising:

analyzing a reduction tree structure in the second gate-level representation of the circuit design.

12. (Currently Amended) ~~The method of Claim 1~~ A method for modeling a circuit design comprising:

synthesizing a circuit design to create a first gate-level representation of the circuit design;

analyzing a second gate-level representation of the circuit design to learn architecture information; and

resynthesizing the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

wherein the analyzing comprises an operator merging learning.

13. (Previously Presented) The method of Claim 12, wherein the operator merging learning comprising:

expressing a complex operation in the first gate-level representation as a summation; and

analyzing a reduction tree structure in the second gate-level representation of the circuit design, the reduction tree corresponds to the complex operation.

14. (Currently Amended) ~~The method of Claim 1~~ A method for modeling a circuit design comprising:

synthesizing a circuit design to create a first gate-level representation of the circuit design;

analyzing a second gate-level representation of the circuit design to learn architecture information; and

resynthesizing the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

~~wherein~~ the analyzing comprising:

identifying a first subcircuit in the first gate-level representation of the circuit design;

identifying a second subcircuit in the second gate-level representation of the circuit design, the second subcircuit corresponding to the first subcircuit; and

calculating a similarity between the first subcircuit and the second subcircuit.

15. (Original) The method of Claim 14, wherein calculating the similarity comprises checking one or more circuit structures.

16. (Original) The method of Claim 14, wherein calculating the similarity comprises checking one or more boolean functions.

17. (Original) The method of Claim 14, wherein calculating the similarity comprises performing one or more simulations.

22. (Currently Amended) ~~The computer-readable storage medium of Claim 21,~~ A computer-readable storage medium having stored thereon computer instructions that, when executed by a computer, cause the computer to:

synthesize a circuit design to create a first gate-level representation of the circuit design;

analyze a second gate-level representation of the circuit design to learn architecture information; and

resynthesize the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

wherein the analyzing comprises a resource sharing learning, and wherein the computer instructions that perform resource sharing learning further comprise computer instructions that, when executed by a computer, cause the computer to:

create one or more resource pairs from sharable resources in the first gate-level representation of the circuit design;

for each of the one or more resource pairs, synthesize a subcircuit that shares the resource pair;

for each of the synthesized subcircuits, calculate a similarity with a corresponding subcircuit in the second gate-level representation of the circuit design; and

identify the synthesized subcircuits having a high similarity with the corresponding subcircuit in the second gate-level representation of the circuit design.

23. (Currently Amended) ~~The computer-readable storage medium of Claim 18~~ A computer-readable storage medium having stored thereon computer instructions that, when executed by a computer, cause the computer to:

synthesize a circuit design to create a first gate-level representation of the circuit design;

analyze a second gate-level representation of the circuit design to learn architecture information; and

resynthesize the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

wherein the analyzing comprises an operator order learning.

24. (Previously Presented) The computer-readable storage medium of Claim 23, wherein the computer instructions that perform operator order learning further comprise computer instructions that, when executed by a computer, cause the computer to:

create one or more operand pairs from operations in the first gate-level representation of the circuit design;

for each of the one or more operand pairs, synthesize a subcircuit for the operand pair;

for each of the synthesized subcircuits, calculate a similarity with a corresponding subcircuit in the second gate-level representation of the circuit design; and

identify the synthesized subcircuits having a high similarity with the corresponding subcircuit in the second gate-level representation of the circuit design.

25. (Original) The computer-readable storage medium of Claim 24, wherein the computer instructions that perform operator order learning further comprise computer instructions that, when executed by a computer, cause the computer to, for each of the synthesized subcircuits having high similarity, create a new operand that signifies the output of the operand pair associated with the synthesized subcircuit.

26. (Currently Amended) ~~The computer-readable storage medium of Claim 18~~ A computer-readable storage medium having stored thereon computer instructions that, when executed by a computer, cause the computer to:

synthesize a circuit design to create a first gate-level representation of the circuit design;

analyze a second gate-level representation of the circuit design to learn architecture information; and

resynthesize the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

wherein the analyzing comprises a multiplier learning.

27. (Previously Presented) The computer-readable storage medium of Claim 26, wherein the computer instructions that perform multiplier learning further comprise computer instructions that, when executed by a computer, cause the computer to:

identify a multiplier in the first gate-level representation of the circuit design;

identify a corresponding multiplier subcircuit in the second gate-level representation of the circuit design;

synthesize the multiplier in the first gate-level representation with a partial product generation implementation; and calculate a similarity for the synthesized partial product generation subcircuit with the partial product generation subcircuit in the corresponding multiplier in the second gate-level representation.

28. (Previously Presented) The computer-readable storage medium of Claim 26, wherein the computer instructions that perform multiplier learning further comprise computer instructions that, when executed by a computer, cause the computer to:

analyze a reduction tree structure in the second gate-level representation of the circuit design.

29. (Currently Amended) ~~The computer-readable storage medium of Claim 18~~ A computer-readable storage medium having stored thereon computer instructions that, when executed by a computer, cause the computer to:

synthesize a circuit design to create a first gate-level representation of the circuit design;

analyze a second gate-level representation of the circuit design to learn architecture information; and

resynthesize the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

wherein the analyzing comprises an operator merging learning.

30. (Previously Presented) The computer-readable storage medium of Claim 29, wherein the computer instructions that perform operator merging learning further comprise computer instructions that, when executed by a computer, cause the computer to:

express a complex operation in the first gate-level representation as a summation; and

analyze a reduction tree structure in the second gate-level representation of the circuit design, the reduction tree corresponds to the complex operation.

31. (Currently Amended) ~~The computer-readable storage medium of Claim 18~~ A computer-readable storage medium having stored thereon computer instructions that, when executed by a computer, cause the computer to:

synthesize a circuit design to create a first gate-level representation of the circuit design;

analyze a second gate-level representation of the circuit design to learn architecture information; and

resynthesize the first gate-level representation of the circuit design to incorporate the learned architecture information from the second gate-level representation of the circuit design,

wherein the computer instructions that perform analyzing further comprise computer instructions that, when executed by a computer, cause the computer to:

identify a first subcircuit in the first gate-level representation of the circuit design;

identify a second subcircuit in the second gate-level representation of the circuit design, the second subcircuit corresponding to the first subcircuit; and

calculate a similarity between the first subcircuit and the second subcircuit.

32. (Original) The computer-readable storage medium of Claim 31, wherein calculating the similarity comprises checking one or more circuit structures.

33. (Original) The computer-readable storage medium of Claim 31, wherein calculating the similarity comprises checking one or more boolean functions.

34. (Original) The computer-readable storage medium of Claim 31, wherein calculating the similarity comprises performing one or more simulations.

35. (Previously Presented) The method of claim 5, wherein the resynthesizing includes:

resynthesizing the first gate-level representation of the circuit design to include the subcircuits identified as having high similarity.

36. (Previously Presented) The method of claim 7, wherein the resynthesizing includes:

resynthesizing the first gate-level representation of the circuit design to include the subcircuits identified as having high similarity.

37. (Previously Presented) The method of claim 10, wherein the resynthesizing includes:

resynthesizing the multiplier in the first gate-level representation to have the partial product generation most similar to the partial product generation in the second gate-level representation.

38. (Previously Presented) The method of claim 11, wherein the resynthesizing includes:

resynthesizing a reduction tree in the first gate-level representation from the reduction tree structure learned from the second gate-level representation.

39. (Previously Presented) The method of claim 13, wherein the resynthesizing includes:

resynthesizing a reduction tree in the first gate-level representation from the reduction tree structure learned from the second gate-level representation.

40. (Previously Presented) The computer-readable storage medium of claim 22, wherein causing the computer to resynthesize includes causing the computer to:

resynthesizing the first gate-level representation of the circuit design to include the subcircuits identified as having high similarity.

41. (Previously Presented) The computer-readable storage medium of claim 24, wherein causing the computer to resynthesize includes causing the computer to:

resynthesizing the first gate-level representation of the circuit design to include the subcircuits identified as having high similarity.

42. (Previously Presented) The computer-readable storage medium of claim 27, wherein causing the computer to resynthesize includes causing the computer to:

resynthesizing the multiplier in the first gate-level representation to have the partial product generation most similar to the partial product generation in the second gate-level representation.

43. (Previously Presented) The computer-readable storage medium of claim 28, wherein causing the computer to resynthesize includes causing the computer to:

resynthesizing a reduction tree in the first gate-level representation from the reduction tree structure learned from the second gate-level representation.

44. (Previously Presented) The computer-readable storage medium of claim 30, wherein causing the computer to resynthesize includes causing the computer to:

resynthesizing a reduction tree in the first gate-level representation from the reduction tree structure learned from the second gate-level representation.